
Cooperative Scheduling for Adaptive Duty Cycling in Asynchronous Sensor Networks

JIANHUI ZHANG¹, ZHI LI¹, FENG XIA², SHAOJIE TANG³, XINGFA SHEN¹ AND BEI ZHAO^{1,4}

¹College of Computer Science and Technology, Hangzhou Dianzi University, 310018 China.

²School of Software, Dalian University of Technology, 116620 China

³College of Science and Technology, Temple University, Philadelphia, PA19122 USA

⁴College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou, 310058 China

Email: jhzhang@ieee.org (Corresponding Author)

To support the sustainable operation of Wireless Sensor Networks (WSNs) using limited energy, duty cycling is a promising solution. However, it is a challenge to guarantee each node communicating with its neighbors under duty cycle when the network is asynchronous. The challenge becomes bigger when nodes' duty cycles are required to be adjusted separately according to their demands in order to save energy and achieve high channel utilization. Existing Low Power Listening (LPL)- and contention-based protocols are not energy-efficient and cannot ensure high channel utility. Additionally, synchronization-based MAC protocols suffer from extra energy consumption and low synchronization precision. This paper proposes a Localized and On-Demand duty cycling scheme (LOD) based on a specifically designed Semi-Quorum System (SQS). LOD can adjust duty cycle of each node adaptively according to its demand so as to avoid channel contention, consequently achieving high channel utilization. This allows the fairness for channel access within asynchronous sensor networks. Extensive experiments are conducted on a real test-bed of 100 TelosB nodes to evaluate the performance of LOD. As compared to B-MAC, LOD substantially reduces contention for channel access and the energy consumption, thus improving the network throughput significantly.

Keywords: Duty Cycling; Semi-Quorum System; Medium Access Control; Wireless Sensor Networks

Received Nov. 30, 2013; revised April 26, 2014

1. INTRODUCTION

Limited energy and channel resources are two main constraints in Wireless Sensor Networks (WSNs). Duty cycling is one of the key ways to save energy and increase channel utilization [1]. In the duty cycled WSNs, one necessary task is to manage the cooperation amongst communicating nodes efficiently. Each sensor node's time is easy to shift because of the imprecise clock. Although there are some existing ways, such as TDMA-based protocols, to cooperate the communication within networks [2][3], they require precise synchronization, which suffers from extra energy cost. The synchronization precision is by no means guaranteed when the network scale is large, such as GreenOrbs [4], because of the various uncontrollable factors in the natural environment [3]. CSMA-based protocol can avoid energy consumption

on synchronization, but it costs much channel resources and energy on the channel contention. This results in low channel utilization. It becomes worse or even infeasible when the duty cycle is extremely low.

Furthermore, sensor nodes must adjust their own duty cycles according to their communication loads in many applications. A proper duty cycle enables each node to finish its communication load sufficiently and can be as low as possible to save energy. It is a challenging problem how to adjust each node's duty cycle adaptively according to its load especially when the whole network is asynchronous and the communication loads among nodes are different.

This paper designs a Localized and On-Demand scheme (LOD) for duty cycle adjustment based on a specifically designed Semi-Quorum System (SQS), denoted by \hat{Q} , based on the Quorum System (QS) [5]. LOD can adjust each node's duty cycle adaptively

according to its load and schedule the active time of every node without adopting any synchronization and channel contention protocol. The key properties of LOD owe to SQS, inheriting the advantages of some QSs with the non-empty intersection and rotation closure properties. QS was applied to establish channel control in dynamic spectrum access networks [6], to save power [7], to maximize throughput in limited information multiparty MAC with QS [8] and to schedule duty-cycling [9]. *The non-empty intersection property* ensures that any pair of neighboring nodes have common active time to communicate with each other. *The rotation closure property* guarantees that any pair of neighboring nodes can have rendezvous active time to communicate with each other without adopting any synchronization protocol within asynchronous networks. However, SQS is more flexible and can ensure each node to be more energy efficient and to have higher channel utilization than QS.

Each node requires its own duty cycle different from others because of its location and task to undertake in its network. So the fixed duty cycle cannot achieve high channel utilization. In many applications, such as canopy closure estimation [4], the duty cycle is simply fixed beforehand or independent among nodes as discussed in Section 1. Compared to the traditional duty cycling protocols, which adjust only the ratio of the active time to the period, LOD can adjust not only the ratio but the slot active mode. LOD can assign each node with a set of active slots. When the ratio is same among nodes, they can wake up at different slots in order to increase the channel utilization and energy efficiency.

LOD can achieve higher channel utilization and fairness for channel access than previous MAC protocols. Energy efficiency under LOD is correspondingly higher than those under other protocols. The detailed contributions of this paper are listed as follows:

- (1) We design the new QS, SQS, which satisfies the nonempty intersection and the rotation closure properties. SQS has high flexible and can obtain lower duty cycle and higher channel utilization and hence save more energy than the QSs under same load.
- (2) Based on SQS, the new protocol, LOD, is designed to adjust the duty cycle of each node adaptively according to its demand. Thus the fairness of channel access and channel utilization are increased compared to existing contention-based MAC protocols.
- (3) LOD guarantees that each pair of neighboring nodes have proper common active time to communicate with each other without synchronization. Therefore, energy and time are saved in comparison with the contention- and synchronization-based MAC protocols.
- (4) A quorum selection method is proposed to allocate quorums for each node so the worst case of channel utilization is bounded and the confliction of active time is decreased or avoided within networks.

- (5) This paper sets up a real test-bed consisting of 100 TelosB nodes, and evaluates the performance of LOD compared to B-MAC.

The organization of this paper is as follows. Section 2 presents the network model and the problem studied in this paper. In Section 3, we introduce the QS technology and present the designing and properties of the new system SQS. Section 4 presents the designing of our scheme LOD and suggests some improvement for it. Meanwhile, the performance of LOD is presented when the certain demand is implemented in asynchronous networks by Section 5. In Section 6, we implement our protocol on the real test-bed with 100 TelosB sensor nodes and analyze the performance of LOD with the experimental results. Section 7 discusses the related works on duty cycling and MAC designing in recent years. The paper is concluded in Section 8.

2. NETWORK MODEL AND PROBLEM STATEMENT

Suppose that the network contains some sensor nodes, $v_i, i = 1, 2, \dots$. Nodes have the same receiving power P_r and the listening power P_l , and different transmission power P_t with others. The interference range is thus different among nodes, and can be described by the interference model, denoted by \mathcal{M} . Each period T contains $|T|$ equivalent slots: $\tau_1, \dots, \tau_{|T|}$. Every node may wake up in a subset of slots to communicate with its neighbors. The subset must contain sufficient common active slots with its neighbors' so that the communication demand among them can be finished. Furthermore, each node's clock is not precise and may result in an unpredictable clock shift δ . The subset of slots has a shift δ accordingly. The challenging problem studied in this paper is how each node wakes up in proper time so that it has sufficient common time with its arbitrary neighbor even when the clock shift exists. Meanwhile, nodes must wake up in as less time as possible to save energy. This paper proposes the new system SQS and designs the new duty cycling scheme LOD so as to save energy and improve the channel utilization.

3. SQS

Before designing SQS, the technology QS [10], denoted by \mathcal{Q} , is firstly introduced.

3.1. Quorum System

DEFINITION 3.1. *Given a period T (a set of slots) to be a universal set, a QS $\mathcal{Q} \subset 2^T$ is a set of subsets of T such that every two subsets intersect with non-empty, i.e. $Q_i \cap Q_j \neq \emptyset$ where $Q_i, Q_j \in \mathcal{Q}$ are called quorums.*

Denote the quorum allocated to node v_i by Q_i , $Q_i \subset \mathcal{Q}$. Figure 1 illustrates a grid QS, denoted by \mathcal{Q}_g , containing two quorums: Q_1 and Q_2 . Each of

TABLE 1. Notation, symbol and abbreviation

| Sym. | Description | Sym. | Description |
|---------------|--------------------|---------------|--------------------|
| T | Period | $ * $ | Set cardinality |
| τ | Time slot | P | Power |
| δ | Clock shift | χ | Color number |
| E | Energy | \mathcal{M} | Interference model |
| \mathcal{D} | Demand | R | SQS row |
| Q | QS | L | SQS column |
| \hat{Q} | SQS | Γ | Rotation operation |
| Q | Quorum | r | # of selected BSQ |
| θ | Neighborhood | l | # of selected NSQ |
| SQS | Semi-Quorum System | | |
| NSQ | Normal Semi-Quorum | | |
| BSQ | Basic Semi-Quorum | | |
| QS | Quorum System | SQ | Semi-Quorum |

them composes of a column and a row. For example, Q_1 composes of the time slots with the black shade in Figure 1. There are some kinds of Qs, and only a few of them, such as the grid, torus and cyclic QS, have the important property: the rotation closure [11].

DEFINITION 3.2. *The rotation of the quorum Q is defined by $\Gamma(Q, \delta) = \{(\tau_j + \delta) \bmod |T| \mid \tau_j \in Q\}$, where δ is a non-negative value and $Q \in \mathcal{Q}$.*

DEFINITION 3.3. *A QS \mathcal{Q} satisfies the rotation closure property in T if any two quorums Q_i and Q_j , $Q_i, Q_j \in \mathcal{Q}$, satisfy the condition: $\forall i \in \{0, \dots, |T|-1\} : Q_i \cap \Gamma(Q_j, \delta) \neq \emptyset$.*

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

 Q_1
 Q_2

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

 Q_2
 Q_3
FIGURE 1. A grid QS \mathcal{Q}_g contains T .

FIGURE 2. An SQS.

3.2. SQS Design

This section designs the new QS, Semi-QS (SQS). The reason to design the SQS is that it can achieve higher energy efficiency than the previous. SQS contains two kinds of semi-quorums: Basic Semi-Quorum (BSQ), and Normal Semi-Quorum (NSQ). It is defined as follows.

DEFINITION 3.4. *Given a period T to be a universal set, an SQS \hat{Q} is a set of subsets of the universal set T and composed of BSQ Q_b , and NSQ Q_n . BSQ and NSQ have non-empty intersection, i.e. $Q_b \cap Q_n \neq \emptyset$, while there is no intersection among BSQs/NSQs.*

BSQ and NSQ are two relative concepts. Let \hat{Q} 's size is $R \times L$. Choose the row in \hat{Q} as BSQ and the

column in \hat{Q} as NSQ. Without loss of generality, let Q_i and Q_j be the i^{th} row and j^{th} column of \hat{Q} respectively, where $1 \leq i \leq R$ and $1 \leq j \leq L$. Thus, we have i^{th} BSQ: $Q_i = \{\tau_k, k = (i-1)L + l, l = 1, \dots, L\}$, and j^{th} NSQ: $Q_j = \{\tau_k, k = j + (m-1)L, m = 1, \dots, R\}$. For example, if SQ Q_1 is a BSQ, then Q_2 and Q_3 are NSQs in Figure 2. Vice versa. Figure 2 shows the SQS constructed based on the grid QS. Notice that SQS can be also designed based on other Qs, such as torus and cyclic QS [11], only if they satisfy the rotation closure property.

This paper takes the grid QS as the basic technology to construct the SQS, and analyzes its properties. The grid-QS based SQS contains $|T|$ slots distributed in a rectangle with $R \times L$ grids. In the grid-QS based SQS, a BSQ is a set of a full row slots while a NSQ is a set of a full column slots. For example, the SQS contains three semi-quorums: Q_1 , Q_2 and Q_3 , where Q_1 is a BSQ and Q_2 and Q_3 are two NSQs in Figure 2. $Q_1 = \{6, 7, 8, 9, 10\}$; $Q_2 = \{2, 7, 12, 17, 22\}$; and $Q_3 = \{4, 9, 14, 19, 24\}$. SQS has the properties including the non-empty intersection and the rotation closure as well. Furthermore, this paper designs the transmutation operation for SQS.

Non-empty Intersection SQS inherits the non-empty intersection property from the previous Qs as Lemma 3.1. According to the method of constructing SQS in this subsection, any two different NSQ/BSQ cannot occupy the same column/row. Lemma 3.2 thus follows.

LEMMA 3.1. *There is at least one rendezvous slot between each pair of NSQ and BSQ in SQS.*

LEMMA 3.2. *The intersection among NSQs/BSQs is empty in SQS.*

Rotation Closure. This important property will be beneficial to design SQS-based protocols in asynchronous networks. With Lemma 3.2, the rotation closure property of SQS is different from the previous Qs.

LEMMA 3.3. *In SQS, any pair of NSQ and BSQ satisfies the rotation closure property.*

Proof. Let \hat{Q} be an SQS. \hat{Q} totally contains $|T|$ elements, which are labeled by some consecutive integers with the row-first manner, such as the example in Figure 2. \hat{Q} 's size is $R \times L$. Choose an arbitrary BSQ Q_i and NSQ Q_j , where $Q_i, Q_j \in \hat{Q}$. Without loss of generality, let Q_i and Q_j be the i^{th} row and j^{th} column of \hat{Q} respectively, where $1 \leq i \leq R$ and $1 \leq j \leq L$. Thus, $Q_i = \{\tau_k, k = (i-1)L + l, l = 1, \dots, L\}$, and $Q_j = \{\tau_k, k = j + (m-1)L, m = 1, \dots, R\}$. There always exist l and m , equaling to j and i respectively, so that $(i-1)L + l = j + (m-1)L$. The above statement proves that $Q_i \cap Q_j \neq \emptyset$.

The rotation of Q_i and Q_j are respectively $\Gamma(Q_i, \delta)$ and $\Gamma(Q_j, \delta)$, where δ is a time value. $\Gamma(Q_i, \delta) =$

$\{(\tau_k + \delta) \bmod |T|, k = (i-1)L + l, l = 1, \dots, L\}$. When $\delta \bmod |T| = 0$, $\Gamma(Q_i, \delta) = Q_i$ and thus $\Gamma(Q_i, \delta) \cap Q_j \neq \emptyset$. When $\delta \bmod |T| \neq 0$, there are two cases: $\delta \bmod L = 0$ and $\delta \bmod L \neq 0$. Under the case that $\delta \bmod L = 0$, $\Gamma(Q_i, \delta)$ moves k rows in parallel, *i.e.* $\Gamma(Q_i, \delta) = Q_{(i+k) \bmod R}$, where k is an integer. Since all BSQs have non-empty intersection with all NSQs, $\Gamma(Q_i, \delta) \cap Q_j \neq \emptyset$. Under the case that $\delta \bmod L \neq 0$, Q_i moves k rows in parallel and l slots, where $l < R$. After moving k rows, Q_i and Q_j have still the non-empty property. Figure 3 shows that Q_i moving l slots results in the case one or the case two. The quorum has still non-empty intersection with any NSQ. Similar proof can be applied to $\Gamma(Q_j, \delta)$. \square

| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

(a) Case one

| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

(b) Case two

FIGURE 3. The quorum moves several rows and slots, and still intersects with any column.

Transmutation Operation Each SQS composes of R BSQs or L NSQs and $R \times L = |T|$. The transmutation operation is defined to adjust R and L while $|T|$ is constant. As another way to adjust duty cycle, it can balance energy consumption among nodes. The network life can be prolonged by leveraging the energy consumption among the whole network. The example in Figure 4 shows how the transmutation operation increases the network life.

| | | | | | |
|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 |

Q_1
 Q_2
 Q_3 Q_4 Q_5

(a) $R \times L = 4 \times 6$

| | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |

Q_1
 Q_2
 Q_3 Q_4 Q_5

(b) $R \times L = 2 \times 12$

| | | |
|----|----|----|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 10 | 11 | 12 |
| 13 | 14 | 15 |
| 16 | 17 | 18 |
| 19 | 20 | 21 |
| 22 | 23 | 24 |

Q_1
 Q_2
 Q_3 Q_4 Q_5

(c) $R \times L = 3 \times 8$

FIGURE 4. By the transmutation operation, a same quorum, such as Q_1 , can have different cardinality.

In each period, the energy consumption of a single node v can be calculated by the below equation.

$$E_T = P_t^v t(\mathcal{D}_v) + P_r t(\mathcal{D}_v) + P_l t_l^v \quad (1)$$

where t_l^v is the total listening time of node v in each period. $t(\mathcal{D}_v)$ denotes the time to finish \mathcal{D}_v .

TABLE 2. Energy consumption of the three SQSs in Figure 4 and the unit is $|\tau|V$

| Node | v_1 | v_2 | v_3 | v_4 | Total energy consumption |
|------------|-------|-------|-------|-------|--------------------------|
| Figure4(a) | 211.2 | 70.4 | 70.4 | 70.4 | 422.4 |
| Figure4(b) | 436.8 | 32.8 | 32.8 | 32.8 | 535.2 |
| Figure4(c) | 98.4 | 145.6 | 145.6 | 145.6 | 535.2 |

The energy consumption of the three SQSs in Figure 4 can be calculated out by Equation (1). Suppose that the current consumption of each sensor node, such as TelosB and MICA2 [12], under receiving, listening and transmitting mode with power $-5dBm$ are $18.8mA$, $18.8mA$ and $14mA$. Suppose that node v_1 has three neighbors v_2 , v_3 and v_4 . v_1 selects two BSQs Q_1 and Q_2 . v_2 , v_3 and v_4 select the NSQs Q_3 , Q_4 and Q_5 respectively. In the common slots between Q_1 , Q_2 and Q_3 , Q_4 , Q_5 , such as slots 1, 2, 3, 13, 14 and 15, v_1 communicates with v_2 , v_3 and v_4 . In other slots (uncommon slots), each node turns its radio into the states: listening or sleeping. Without loss of generality, we suppose the four nodes spend one half common time to receive and other half common time to transmit. In Figure 4, the energy consumption of the three SQSs is illustrated in Table 2.

Define the network life as the duration from the network start moment to the first node using up its energy. In Table 2, the total energy consumption in the SQS of Figure 4(a) is minimal but the network lifetime is $\min\{\frac{1}{211.2}, \frac{1}{70.4}, \frac{1}{70.4}, \frac{1}{70.4}\} = \frac{1}{211.2}$ if the total energy that each node initially has is one τV . In the SQS of Figure 4(b), the energy consumption of v_2 , v_3 and v_4 is minimized but the network lifetime is $\min\{\frac{1}{98.4}, \frac{1}{145.6}, \frac{1}{145.6}, \frac{1}{145.6}\} = \frac{1}{145.6}$. In the SQS of Figure 4(c), the energy consumption of all nodes is not minimal comparing to those in Figure 4(a) and 4(b). The total energy consumption is not minimal as well. But the network lifetime is $\min\{\frac{1}{436.8}, \frac{1}{32.8}, \frac{1}{32.8}, \frac{1}{32.8}\} = \frac{1}{436.8}$, and maximal among all SQSs in Figure 4. Thus the energy minimization does not necessarily result in maximizing the network life. Although the SQSs in Figure 4(b) and Figure 4(c) consumes the same energy totally, the later has better energy consumption balance. The example explores the hint that the network life can be prolonged by balancing the energy consumption among nodes.

4. LOD

This section designs the new method LOD to let each node locally decide its duty cycle by cooperating with its neighbors when nodes are asynchronous.

4.1. Designing LOD

LOD composes of three steps: period assignment, tree construction and duty cycle designing.

Step I: Assign each node with a period by the vertex coloring. By the distributive coloring algorithm [13], each node is assigned a color. The minimal number of colors to color all nodes is denoted by χ , which depends on the interference model \mathcal{M} . Assigned a period T_k to v_i with color k , $k = 1, \dots, \chi$.

Step II: Each node adjusts its duty cycle according to its demand. Each node v_i collects its neighbor v_j 's ID, $v_j \in \theta(v_i)$. v_i inquiries its neighbors' demands. The demand composes of transmitting or receiving data. Denote the data transmitted from v_i to v_j by \mathcal{D}_{ij} .

- v_i calculates the time to finish the demands respectively for all neighbors;
- v_i selects r_i BSQs for itself and l_j NSQs respectively for its neighbor v_j ;

Step III: Each node implements the transmutation operation.

- v_i determines the new size of SQS by calculating the number of NSQs;
- v_i sends the number of NSQs and the size of SQS to its neighbors.

Notice that each node is assigned a period in the step I. In the period, the node assigns itself some BSQs and some NSQs to its neighbors. Thus each node can wake up in its own and neighbors' periods. The below context explains the detailed implementation of the above three steps and determines the values of the variables in these steps.

Demand Time. In the step II, each node's active time of depends on not only its demand but also other factors, such as the data rate. The data rate is calculated by averaging the data delivering over a duration so that variety of factors affecting the time to implement can be included. Recall that the demand consists of two types: data transmitting and receiving. v_i thus adjusts its active time to be at least $t(\mathcal{D}_{ij} + \mathcal{D}_{ji})$. v_i may have several neighbors, its active time should be the sum of the time to implement all the demands between it and its neighbors. Like lots of previous works [14][15][16], nodes can exchange their demands by some beacons at the initial phase of the network life or the beginning of each period under LOD.

Duty Cycling. Step II wakes up each node in certain amount of quorums so that the demand can be implemented. The energy consumption in each period can be represented by two parts: E_i consumed by v_i and E_j consumed by v_i 's neighbors v_j , $v_j \in \theta(v_i)$. The energy consumption E_i is described in Equation (2).

$$E_i = P_t^i \sum_{v_j \in \theta(v_i)} t(\mathcal{D}_{ij}) + P_r \sum_{v_j \in \theta(v_i)} t(\mathcal{D}_{ji}) + P_l t_l^i \quad (2)$$

where t_l^i is v_i 's listening time. The energy consumption E_j of the neighbor v_j is described in Equation (3).

$$E_j = P_t^j t(\mathcal{D}_{ji}) + P_r t(\mathcal{D}_{ij}) + P_l t_l^j \quad (3)$$

Suppose that v_i selects r_i BSQs and v_j selects l_j NSQs. v_i should have sufficient common slots with

its neighbors to implement the demands among them. The the number of common slot should be at least $\frac{1}{|\tau|} t(\mathcal{D}_{ij} + \mathcal{D}_{ji})$, where $|\tau|$ is the size of slot τ , such as one second or millisecond. In order to save energy, there should be $r_i \times l_j = \frac{1}{|\tau|} t(\mathcal{D}_{ij} + \mathcal{D}_{ji})$. On the other hand, each node should spend energy on the listening at those uncommon slots. For example, when a node selects the quorum Q_3 in Figure 4(a), it should listen at the slots 13 and 19. Therefore, the node v_i and its neighbor v_j have the listening time as described respectively in Equation (4) and (5).

$$t_l^i = |\tau| (r_i L_i - \sum_{v_j \in \theta(v_i)} r_j l_j) \quad (4)$$

$$t_l^j = |\tau| l_j (R_i - r_j) \quad (5)$$

where neighbors have different demand and thus select their own row amount r_j . v_i has to select sufficient number of BSQs to meet the all demands of its neighbors so $r_i = \max_{v_j \in \theta(v_i)} r_j$.

In Step II, each node v_i selects BSQs for its own and NSQs for its neighbors so that the overall energy consumption can be reduced. The overall energy consumption of v_i and its neighbor can be obtained by adding two sides of Equation (2) and (3) respectively. Notice that the time to implement the demand $t(\mathcal{D}_{ij} + \mathcal{D}_{ji})$ should be satisfied. Equation (2) and (3) indicates that the only way to change the energy consumption is to adjust the listening time. It thus can minimize the energy consumption in Equation (2) and (3) in order to save the overall energy. Since the listening power is same for all nodes, to minimize the overall energy consumption is equivalent to minimizing the listening time.

$$\min t_l^i + \sum_{v_j \in \theta(v_i)} t_l^j \quad (6)$$

The solution to the above equation can be easily obtained by follows. According to Equation (4) and (5), we can have the below expression.

$$\begin{aligned} t_l^i + \sum_{v_j \in \theta(v_i)} t_l^j &= |\tau| (r_i L_i + \sum_{v_j \in \theta(v_i)} l_j R_i - 2 \sum_{v_j \in \theta(v_i)} r_j l_j) \\ &= |\tau| r_i L_i + R_i \sum_{v_j \in \theta(v_i)} \frac{t(\mathcal{D}_{ij} + \mathcal{D}_{ji})}{r_j} \\ &\quad - 2 \sum_{v_j \in \theta(v_i)} t(\mathcal{D}_{ij} + \mathcal{D}_{ji}) \\ &\geq |\tau| r_i L_i + R_i |\theta(v_i)| \frac{t(\mathcal{D}_{ij} + \mathcal{D}_{ji})}{r_i} \\ &\quad - 2 \sum_{v_j \in \theta(v_i)} t(\mathcal{D}_{ij} + \mathcal{D}_{ji}) \\ &\geq 2\sqrt{|T||\theta(v_i)|t(\mathcal{D}_{ij} + \mathcal{D}_{ji})} \\ &\quad - 2 \sum_{v_j \in \theta(v_i)} t(\mathcal{D}_{ij} + \mathcal{D}_{ji}) \end{aligned} \quad (7)$$

where the second equality is because $r_i \times l_j = \frac{1}{|\tau|} t(\mathcal{D}_{ij} + \mathcal{D}_{ji})$. The first inequality is because $r_i = \max_{v_j \in \theta(v_i)} r_j$.

The second inequality is because the Cauchy inequality and $|\tau|L_iR_i = |T|$. Thus, the minimal energy consumption in (6) can be obtained when the condition $r_i = \sqrt{(|\theta(v_i)|R_i t(\mathcal{D}_{ij} + \mathcal{D}_{ji})) / (|\tau|L_i)}$ is satisfied.

We can check the correctness of the condition. In the example of Figure 4(a), $R_i = 4$, $L_i = 6$, $|\theta(v_i)| = 3$ and the slot amount to implement is $t(\mathcal{D}_{ij} + \mathcal{D}_{ji})|\tau| = 2$ so $r_i = \sqrt{3 * 4 * 2 / 6} = 2$. By Step II, we can adjust nodes' duty cycled so that the overall energy consumption is minimized. Table 2 has already proved that the energy consumption in the example of Figure 4(a) is minimal among all the example in Figure 4. Notice that we can induce that $\frac{r_i}{|\theta(v_i)|l_j} = \frac{R_i}{L_i}$ from the condition. In the equality, the ratio between the numbers of selected BSQs and NSQs is proportional to the ratio of the row amount to the column amount of v_i 's SQS. The equality hints an explicit way to find the optimal solution to (6).

The remaining question is how to select NSQs for each neighbor. The key metric for NSQs selection is still energy saving. Recall that the condition enabling the first inequality correct in Equation (7) is that $r_i = \max_{v_j \in \theta(v_i)} r_j$. When all neighbors need the same time to implement their demands, the first inequality in Equation (7) becomes equality and thus the listening time in Equation (7) can be minimal. However, neighbors may need different time to implement their demands so the way to select NSQs affects the energy consumption. Figure 5 gives an example in which node v_i has two neighbors v_j and v_k with different demand. Suppose that v_j needs two slots to implement its demand while v_k needs four slots. Under the method one in Figure 5(a), v_j selects the NSQ Q_5 and v_k selects the NSQ Q_6 . v_k needs only the slots, 26 and 32 to listen. v_i has to select four BSQs: Q_1 , Q_2 , Q_3 , and Q_4 . Under the method two in Figure 5(b), v_j selects the NSQ Q_5 and v_k selects the NSQs Q_6 and Q_7 . v_k has to wake up to listen at the slots: 14, 15, 20, 21, 26, 27, 32 and 33. v_i need only select two BSQs: Q_1 and Q_2 . v_i need not wake up to listen at the slots: 3, 9, 13, 15, 16, 17, 18, 19, 21, 22, 23 and 24. Compared to the method one, v_k needs six slots more to listen and v_i saves twelve slots to listen. Obviously, the method two saves energy much more than the method one. Actually, it can save more energy to minimize the number of BSQs selected by v_i according to the condition to satisfy the first inequality in Equation (7). So the rule to select NSQs for each neighbor follows: (1) Suppose that the neighbor v_j needs minimal time to implement its demand among all neighbors. Select one NSQ for v_j . Denote the number of its rows in the NSQ to implement its demand by r_j . (2) Select $\lceil \frac{t_k}{r_j} \rceil$ NSQs for other neighbor v_k , where t_k is the time that v_k needs.

Recall that the network life is defined as the time

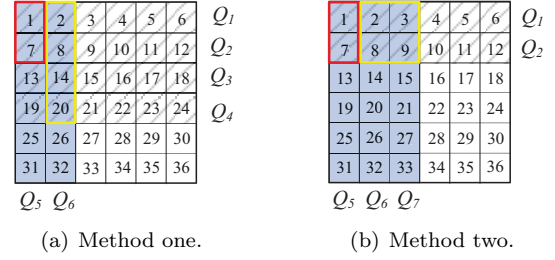


FIGURE 5. The way to select NSQs for neighbors affect energy consumption.

up to the first node using up its energy. The energy consumption balance is thus very important. Step III is to implement the transmutation operation so that the energy consumption can be balanced between each node and its neighbors. After each node selects its BSQ and the NSQs for its neighbors, the variables r and l are determined in Equation (4) and (5). Recall that the active time $t(\mathcal{D}_{ij} + \mathcal{D}_{ji})$ should be satisfied to implement the demand. So only the listening time can be adjusted. Usually, node v_i consumes energy much faster than its neighbors $\theta(v_i)$ since it has often more than one neighbors. We expect all nodes consume energy in almost the same speed. The easy way is to choose the neighbor v_k from $\theta(v_i)$, whose energy consumption is middle in $\theta(v_i)$, and then to reduce the energy consumption difference between v_i and v_k . According to Equation (4) and (5), the row of the SQS should be adjusted by the following equation.

$$L_i(new) = \max\{L_i(old) - \lceil \frac{E_i - E_k}{2P_l|\tau|r_i} \rceil, \sum_{v_j \in \theta(v_i)} l_j\} \quad (8)$$

where $L_i(old)$ and $L_i(new)$ are respectively the column amount in the SQS before and after the transmutation operation. By the above equation, v_i adjusts the column amount $L_i(old)$ and obtains $L_i(new)$ so that its energy consumption can be close to the node v_k 's, whose energy consumption is middle among v_i 's neighbors.

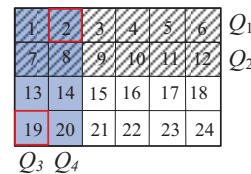


FIGURE 6. Overlap confliction.

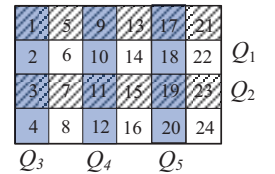


FIGURE 7. Dispersing quorums.

4.2. Dispersing Quorums for Asynchronization

As stated in Lemma 3.3, the rotation closure property of SQS guarantees the non-empty intersection between BSQs and NSQs when the network is asynchronous. When a node's work load is not too high to overcrowd

the SQs in its period, the quorum dispersing can avoid or decrease the overlap among SQs caused by the clock shift. For example, the Q_3 overlaps with Q_4 on the slot 2 because of the clock shift in Figure 6. Both of Q_3 and Q_4 are active at the slot 2 so the confliction happens.

The clocks of the sensor nodes, such as TelosB and MicaZ, are often imprecise but the clock shift is limited [17]. Elson *et al.* tested the Berkeley sensor node [18] based on TinyOS and obtained the distribution of the receivers' clock shifts, which obey the Gaussian distribution with zero expectation and the variance $11.1\mu sec$ under confidence 99.8% [17]. Knowing the clock shift limitation, the overlap confliction among SQs can be avoided or decreased by properly dispersing the SQs. Set the variable δ to be 3 times of the variance, *i.e.* $\delta = 33.3\mu sec$. We disperse the quorums including BSQs and NSQs apart from each other with the interval δ as shown in Figure 7. These quorums then do not overlap with each other with high probability. Since NSQs are assigned to different neighbors, there should be $f(\delta)$ NSQs interval between the adjacent NSQs selected by different neighbors as shown in Figure 7. $f(\delta)$ is determined by Equation (9).

$$f(\delta) = \begin{cases} \lfloor \frac{\delta}{\tau R} \rfloor, & \frac{\delta}{\tau R} - \lfloor \frac{\delta}{\tau R} \rfloor < R_i - r_i \\ \lceil \frac{\delta}{\tau R} \rceil, & otherwise \end{cases} \quad (9)$$

4.3. Energy Efficiency Improvement

This block provides two methods to improve the energy efficiency for each node. They are the quorum tessellating and shrink. In the quorum tessellating, those active slots are tessellated into new SQs as show in Figure 8. Recall that each node has to be active at each slot in its quorum even when it is listening. By the quorum tessellating, much energy can be saved. In Figure 8, the slot 30 in the original SQS is tessellated into a new refined SQS. The properties of the non-empty intersection and rotation closure are still hold after the quorum tessellating. After one generation of QS tessellating, the saved time not to wake up in the example of Figure 8 can be easily calculated as follows. The size of a BSQ in the original SQS is R , and then the BSQ can save energy $R(L - \frac{1}{L})$ by the quorum tessellating.

In the method of the quorum shrink, each node need wake up at part slots of its quorum. The real sensor nodes, such as TelosB and MicaZ, have limited clock shift as $11.1\mu sec$ under confidence 99.8% [17]. So it need only wake up around the intersected slot. For example, the slot size is $11.1\mu sec$ in Figure 9, and v_i selects the quorum Q_1 . v_i can have high probability to connect with other node selecting the quorum Q_2 when it wakes up only at slots 6, 7 and 8. This method can greatly save energy and increase the channel utilization.

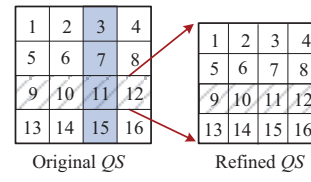


FIGURE 8. Quorum tessellating.

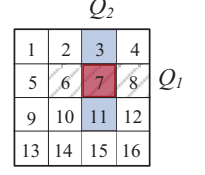


FIGURE 9. Quorum shrink.

5. PROPERTIES OF LOD

In the previous sections, SQS enables LOD to have two properties: the non-empty intersection and the rotation closure. LOD can establish connection among nodes even when they are not synchronized. It can adjust each node's duty cycle so that its demand can be implemented and energy can be saved and the energy consumption can be balanced in the whole network. This section analyzes another properties of LOD: the maximal SQS load and channel utility. The maximal SQS load is referred as the maximal load that each node can afford of per period under the LOD.

The maximal SQS load. The maximal load that each node can afford of is affected by the interference models \mathcal{M} , the SQS structure and duty cycling. In each period, the time to implement node v_i 's demand is $\sum_{v_j \in \theta(v_i)} t(\mathcal{D}_{ij} + \mathcal{D}_{ji})$, which is determined by the

data rate. Section 4.2 indicates that there are $f(\delta)$ NSQs between adjacent NSQs in order to overcome the negative affection of asynchronization. Since there are $|\theta|$ neighbors and each neighbor selects l_j NSQs, $\sum_{v_j \in \theta(v_i)} (l_j + f(\delta)) \leq L_i$. There are at most

$(L_i - |\theta(v_i)|f(\delta))R_i\tau$ time to implement demands of all neighbors in each period. Recall that it needs χ colors to color all nodes under the interference model \mathcal{M} . Each color is assigned a period so there are totally χ periods. That means each node's period can wake up in every χ periods. Therefore, the maximal demand that each node can afford of per period is $\sum_{v_j \in \theta(v_i)} (\mathcal{D}_{ij} + \mathcal{D}_{ji}) \leq$

$\frac{\rho}{\chi}(L_i - |\theta(v_i)|f(\delta))R_i\tau$, where ρ is the data rate. When the network works under (ultra) low duty cycle, *i.e.* $\frac{r_v\tau + \delta}{\tau R} < 1$, $f(\delta) = 0$. So $\sum_{v_j \in \theta(v_i)} (\mathcal{D}_{ij} + \mathcal{D}_{ji}) \leq \frac{\rho T}{\chi}$. It

means the asynchronization would have no effect on the maximal demand that each node can implement when the network works under (ultra) low duty cycle.

Channel Utilization. Channel utilization is a popular metric to measure MAC protocols' performance [16]. Rhee *et al.* gave an equation to calculate the channel utilizations under B-MAC, PTDMA and Z-MAC as Equation (10) [19].

$$\mathcal{U} = \frac{t_c}{t_c + t_l + t_s} \quad (10)$$

where t_c is the communication time including transmit-

ting and receiving data. t_l and t_s are the listening and sleeping time respectively. Under the LOD, the channel is utilized in the common active slots between BSQs and NSQs, which includes the time to transmit and receive data. LOD wakes nodes up in some additional slots so that the rotation closure property can be satisfied. The utilization under LOD can be described by the followed equation.

$$\mathcal{U} = \frac{l_i r_j}{\chi |T|} \quad (11)$$

where l_i and r_j are selected sufficient to implement the node v_i 's demand. On one hand, the channel utilization can be adjusted by changing the cardinality T of the period. On other hand, it is also related to the demand in each period. SQS has higher channel utilization than the Grid QS. For example, nodes should be active at total 16 slots in Figure 1 in order to have two common active slots. Meanwhile, nodes need only be active at total 13 slots in Figure 2.

6. EXPERIMENTAL RESULTS

This section evaluates the performance of LOD and compares it with B-MAC on a real test-bed, which consists of 100 TelosB sensor nodes as shown in Figure 10. Each node runs TinyOS. We compare the performance of LOD with B-MAC on the network throughput, Packet Reception Ratio (PRR) and the energy consumption.



FIGURE 10. Our WSNs testbed with 120 nodes.

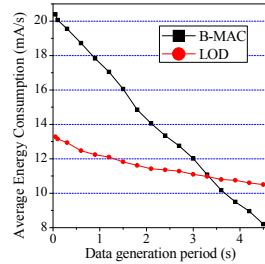


FIGURE 11. Energy Consumption Comparison between LOD and B-MAC.

6.1. Experimental Setup

The experiment takes the data aggregation as the application example so a Minimum Spanning Tree (MST) is constructed based on all nodes by the existing method: BFS. The experiments compose of two phases. In the first phase, LOD is implemented. In the second phase, the MST is constructed and rooted at the sink. The data aggregation is implemented under LOD. In the tree, all leaf nodes are not assigned periods and share the periods with their parent nodes. Meanwhile, the protocol B-MAC is also implemented based on the MST. The reason to adopt B-MAC is among the

first random access MAC protocols to adopt the LPL technique, which is a feasible way to deal with the asynchronous communication among nodes. Since B-MAC has no ability of duty cycling, the duty cycle is set as 20% in the experiment for B-MAC.

Under LOD, each \mathcal{Q} contains 100 time slots, *i.e.* $|T| = 100$. Each slot is respectively set as 50ms, 1s, 2s and 5s. Each node samples data in every 100ms, 200ms, 300ms, 500ms, 800ms, 1s, 1.5s and 2s, which is called as the data generation period in Figure 12 and 13.

6.2. Performance Comparison

The evaluation of LOD and B-MAC on the network capacity and PRR illustrate their performance on channel utilization and fairness indirectly. Energy consumption is also measured.

Throughput. Figure 12 shows the network throughput under LOD and B-MAC respectively. Each node generates data at different rates. Since all nodes must compete for channel access in B-MAC, much channel resources is wasted when delivering data packets. Thus B-MAC achieves a lower network capacity than LOD, as shown in Figure 12. When the sampling period is small, such as 100ms, 200ms and 300ms, the throughput under B-MAC is much lower than that under LOD. Because packets are transmitted more frequently under smaller sampling period, B-MAC observes a greater degree of channel contention. Although the throughput under LOD is higher than B-MAC, it is actually quite low because the maximal data rate of TelosB node can reach 250Kbps. One of main reasons is that asynchronous clocks make quorums overlap with each other and each node's clock shift is arbitrary and unknown in LOD. Thus connotative channel contention cannot be avoided. When the size of slot is smaller, the possibility that quorums overlap is higher. Therefore, the throughput under both B-MAC and LOD is close to each other as shown in Figure 12(a).

PRR. The PRR reflects the channel utility in the network. The PRRs under both B-MAC and LOD are shown in Figure 13. The PRR under B-MAC increases with the data generation period. But PRR under LOD keeps above 80% under almost all data generation periods. The slots size has much effect on the throughput under B-MAC instead of that under LOD. When the size of slot is smaller, such as, 50ms, PRR under LOD decreases much, and is lower than that under B-MAC when data generation is higher than 900ms as shown in Figure 13(a). The reasons are similar to those given for the network throughput.

Energy consumption. The experimental results on energy consumption are shown in Figure 11 when the slot size sets to 1s. In this figure, the energy consumption is the average value per node per second. The energy consumption under LOD keeps about 6mA/s per nodes in different data generation periods, while under B-MAC varies from more than

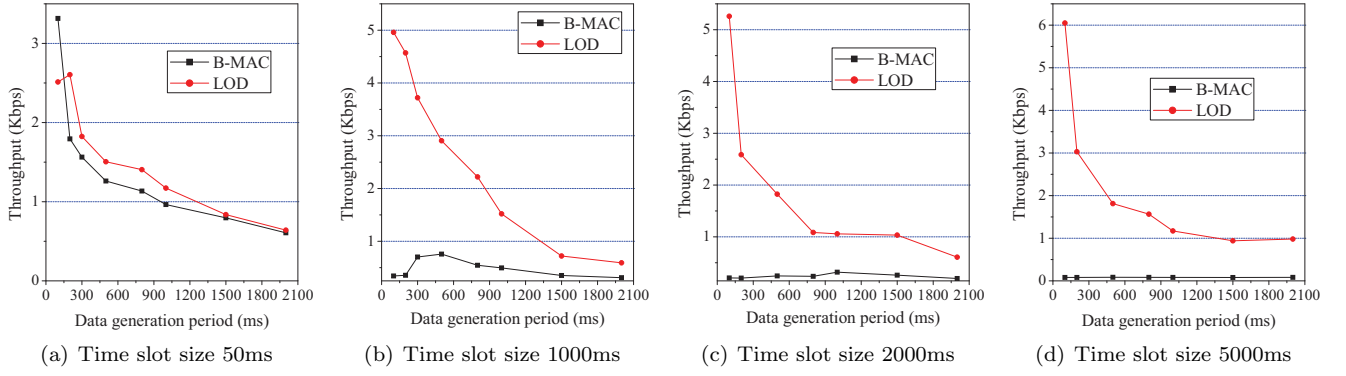


FIGURE 12. The network throughput respectively under LOD and B-MAC with different data generation intervals.

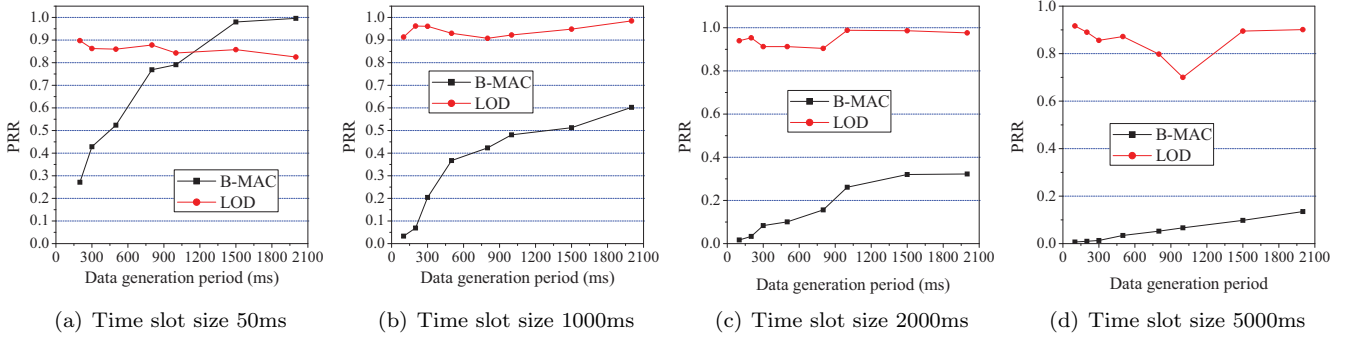


FIGURE 13. The network PRR respectively under LOD and B-MAC with different data generation periods.

20 mA/s to 8mA/s. It is obvious that energy consumption under B-MAC is greatly affected by the data generation periods because shorter data generation periods cause more contention for channel under B-MAC. The energy consumption in Figure 11 indicates that LOD can decrease the channel contention dramatically. Furthermore, we can notice that the energy consumption of B-MAC is lower than that of LOD when the data generation period is over 3.3 s. The reasons are twofold. Firstly, the channel contention decreases. Secondly, nodes lose more access to communicate with their neighbors because of the shift of their clocks.

7. RELATED WORK

7.1. Duty Cycle

In WSNs, operation at a certain duty cycle can bridge the gap between the limited energy and longing network life [20]. Gu and He designed a data forwarding technique to optimize the data delivery ration, end-to-end delay or energy consumption in low-duty-cycle sensor networks under a synchronized mode [1]. Guo and Gu *et al.* designed an Opportunistic Flooding scheme for low-duty-cycle networks with unreliable wireless links and predetermined wording schedules when the network is locally synchronized [21]. Wand and Liu provided a benchmark for assessing diverse duty-cycle-aware broadcast strategies and extended it

to distributed implementation [22]. It translated the broadcast problem into a graph equivalence in order to seek a balance between efficiency and latency with coverage guarantees.

Since the clock is easy to shift in WSNs, some works considered network operation under asynchronous clock. One of most famous techniques to establish communication among nodes under asynchronization is the LPL technique [16]. Under this technique, some preambles are firstly sent before the data so that nodes can communicate with each other while the clock shift exists. The LPL technique cannot work well when the clock shift is larger than the overall length of all the preambles. Sun and Johnson *et al.* designed an asynchronous duty-cycled broadcast, under which, nodes may be active for very long time when they need broadcast their data to a large number of neighbors, which awake up in different time [23].

There are also many other works studying the network performance under duty cycled networks, such as latency [24][25][26], opportunistic data aggregation [27], and reliable data delivery [28][29]. Nath and Gibbons analyzed the performance of geographic routing over duty-cycled nodes and presented a sleeping scheduling algorithm that can be tuned to achieve an expected routing latency [30] and coverage [30]. After these works, researchers found that the network traffic or the demand has great effect on the duty cycle. Lee and Choi *et al.* proposed a traffic-adaptive MAC: AMAC, to ad-

just the duty cycle of each node according to its traffic but it did not consider the synchronization among nodes and the feasibility of the traffic-based duty cycle adjustment since WSNs often affords of different kinds of tasks and network traffic is variable [31].

7.2. MAC protocol

In WSNs, MAC is one main class of protocols considering the node sleep/wakeup or duty cycle adjustment. Their goals are to save energy, to improve the network throughput and/or to shorten the transmission delay. They can be classified into two classes: synchronization based and asynchronization based. Synchronization based protocols include S-MAC [32], T-MAC [33], U-MAC [34], P-MAC [35] and H-MAC [36]. Asynchronization based ones include D-MAC [37], B-MAC [16], Wise-MAC [38], SyncWUF [39] and ACDA [40]. Paper [40] offered the classification and summarization of the above protocols.

Some protocols are designed to combine the advantages of TDMA and CSMA. Rhee and Warrier *et al.* proposed a hybrid MAC protocol, called Z-MAC [19]. In Z-MAC, a node always performs carrier-sensing before it transmits during a slot. Thus Z-MAC consumes much energy on the carrier-sensing. Z-MAC also requires local synchronization among senders in two-hop neighborhoods. S-MAC [32] and T-MAC [33] are also a hybrid of CSMA and TDMA and employ RTS/CTS mechanism to solve synchronization failure. Since these protocols use RTS/CTS, their overhead is quite high [19]. B-MAC [16] is the default MAC in the operating system of Mica2 and adopts LPL to solve the asynchronization. Since LPL consumes much energy, X-MAC reduces the energy consumption and latency by employing short preamble and embedding address information of the target in the preamble [41]. Thus the non-target receivers can quickly go back to sleep and the energy is saved. LPL based preamble transmission may occupy the medium for much longer than actual data transmission. So [42] designed an asynchronous duty cycle MAC: RI-MAC. In RI-MAC, the energy would be wasted especially when the traffic load is low and the interference would be increased since each node should broadcast a beacon periodically no matter the sender has data to transmit or not.

MAC protocols are also designed to reduce energy consumption, such as S-MAC [32] and T-MAC [33]. Zheng and Hou *et al.* considered LPL approaches, such as WiseMAC and B-MAC, are limited to duty cycles of 1-2% and designed a new MAC protocol called scheduled channel polling (SCP) to ensure that duty cycles of 0.1% and below are possible [14]. It dynamically adjusts duty cycles in the face of busy networks and streaming traffic in order to reduce the latency.

Sha and Xing *et al.* presented a MAC protocol, C-MAC to achieve high-throughput bulk communication

for data intensive sensing applications [43], but it did not consider duty cycle. Kim and Shin *et al.* proposed a lighted-weight channel hopping mechanism, thus avoiding redundant channel assignment by not allocating fixed channels to the nodes [44]. Synchronization was also implemented by initial time synchronization, error compensation and slot assignment and retrieval. However, message was cost in this process and duty cycle was not considered in the protocol.

Bian *et al.* applied QS to establish channel control in a dynamic spectrum access networks [6]. Wu *et al.* used it save power [7]. Chaporkar *et al.* maximized throughput in limited information multiparty MAC with QS [8]. Lai and Ravindran presented quorum-based duty-cycling schedule where nodes send a beacon message at the beginning of wake-up slots [9].

MAC mainly allocates channel resources for these conflicting neighbors. Some of them considered the communication among nodes when their clocks were asynchronous. However, they still need synchronization protocols to cooperate the communication time, or add extra preamble to overcome the clock shift, such as LPL. This paper argues the new method LOD to ensure the communication among nodes without using any synchronization protocols in case of imprecise clock.

8. CONCLUSION

This paper designed the SQS based on QS. Based on SQS, we proposed the localized scheme, LOD, to adjust nodes' duty cycle adaptively according to its demand such that they can fairly use the channel. LOD combines the advantages of both TDMA and CSMA and doesn't need extra synchronization protocol. When the network is asynchronous, LOD can still guarantee that any pair of neighboring nodes has sufficient common active time to communicate with each other. LOD can increase the channel utilization by adjusting each node's duty cycle based on its demand. We established the WSNs testbed with 100 TelosB nodes and evaluated the performance of LOD. Compared to B-MAC, LOD significantly improves the performance such as network throughput and PRR. It is still a challenging problem to design energy-efficient protocols to establish communication in the duty-cycled networks when the clock is imprecise, especially when the duty cycle is ultra-low.

ACKNOWLEDGEMENT

This work is supported by the Zhejiang Province Natural Science Foundation under Grants No.Y14F020171, the Major Program of the National Natural Science Foundation of China (NSFC) under Grants No.61190113, the NSFC under Grant No.61003298, and the open funding of Science and Technology on Sonar Laboratory in Hangzhou City under Grant No.KF201103.

REFERENCES

- [1] Gu, Y. and He, T. (2007) Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links. *ACM Proceedings of the 5th international conference on Embedded networked sensor systems (Sensys)*, Sydney, Australia, Nov.6-9, pp. 321–334. ACM, New York, NY, USA.
- [2] Sivrikaya, F. and Yener, B. (2004) Time synchronization in sensor networks: a survey. *IEEE Network*, **18**, 45–50.
- [3] Schmid, T., Charbiwala, Z., Anagnostopoulou, Z., Srivastava, M. B., and Dutta, P. (2010) A case against routing-integrated time synchronization. *ACM Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Zürich, Switzerland, Nov.3-5, pp. 267–280. ACM, New York, NY, USA.
- [4] Mo, L., He, Y., Liu, Y., Zhao, J., Tang, S., Li, X., and Dai, G. (2009) Canopy closure estimates with greenorbs: Sustainable sensing in the forest. *ACM Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Berkeley, California, Nov.3-6, pp. 99–112. ACM, New York, NY, USA.
- [5] Malkhi, D. and Reiter, M. (1998) Byzantine quorum systems. *Springer Distributed Computing*, **11**, 203–213.
- [6] Bian, K., Park, J., and Chen, R. (2009) A quorum-based framework for establishing control channels in dynamic spectrum access networks. *ACM The Annual International Conference on Mobile Computing and Networking (MobiCom)*, Beijing, China, Sept.20-25, pp. 25–36. ACM, New York, NY, USA.
- [7] Wu, S., Chen, M., and Chen, C. (2008) Fully adaptive power saving protocols for ad hoc networks using the Hyper Quorum System. *IEEE The 28th International Conference on Distributed Computing Systems (ICDCS)*, Beijing, China, June 17-20, pp. 785–792. IEEE, New Brunswick, NJ USA.
- [8] Chaporkar, P., Sarkar, S., and Shetty, R. (2006) Dynamic quorum policy for maximizing throughput in limited information multiparty mac. *IEEE/ACM Transactions on Networking (TON)*, **14**, 835–848.
- [9] Lai, S. and Ravindran, B. (2010) On multihop broadcast over adaptively duty-cycled wireless sensor networks. *The 6th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, California, USA, June 21-23, pp. 158–171. IEEE, New Brunswick, NJ USA.
- [10] Tseng, Y., Hsu, C., and Hsieh, T. (2003) Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. *Elsevier Computer Networks*, **43**, 317–337.
- [11] Jiang, J., Tseng, Y., Hsu, C., and Lai, T. (2005) Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks. *Springer Mobile Networks and Applications*, **10**, 169–181.
- [12] MEMSIC Inc. (2009). Telosb mote platform datasheet. <http://www.memsic.com/>.
- [13] Panconesi, A. and Srinivasan, A. (1992) Improved distributed algorithms for coloring and network decomposition problems. *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing (STOC)*, Victoria, British Columbia, Canada, May 4-6, pp. 581–592. ACM, New York, NY, USA.
- [14] Zheng, R., Hou, J., and Sha, L. (2003) Asynchronous wakeup for ad hoc networks. *The fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Annapolis, MD, USA, June 1-3, pp. 35–45. ACM, New York, NY, USA.
- [15] Dutta, P. and Culler, D. (2008) Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. *ACM Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Raleigh, NC, USA, Nov.5-7, pp. 71–84. ACM, New York, NY, USA.
- [16] Polastre, J., Hill, J., and Culler, D. (2004) Versatile low power media access for wireless sensor networks. *ACM Proceedings of the 2th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Baltimore, Maryland, Nov.3-5, pp. 3–5. ACM, New York, NY, USA.
- [17] Elson, J., Girod, L., and Estrin, D. (2002) Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, **36**, 147–163.
- [18] Kahn, J., Katz, R., and Pister, K. (1999) Next century challenges: mobile networking for “Smart Dust”. *ACM/IEEE The Annual International Conference on Mobile Computing and Networking (MobiCom)*, Washington, USA, August 15-20, pp. 271–278. ACM, New York, NY, USA.
- [19] Rhee, I., Warrier, A., Aia, M., Min, J., and Sichitiu, M. (2008) Z-MAC: a hybrid MAC for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, **16**, 511–524.
- [20] Zhang, J., Tang, S., Shen, X., Dai, G., and Nayak, A. (2011) Quorum-based localized scheme for duty cycling in asynchronous sensor networks. *IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Valencia, Spain, October 17-22, pp. 440–449. IEEE, New Brunswick, NJ USA.
- [21] Guo, S., Gu, Y., Jiang, B., and He, T. (2009) Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. *ACM The Annual International Conference on Mobile Computing and Networking (MobiCom)*, Beijing, China, Sept.20-25, pp. 133–144. ACM, New York, NY, USA.
- [22] Wang, F. and Liu, J. (2009) Duty-cycle-aware broadcast in wireless sensor networks. *The 28th Annual IEEE International Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 19-25, pp. 468–476. IEEE, New Brunswick, NJ USA.
- [23] Sun, Y., Gurewitz, O., Du, S., Tang, L., and Johnson, D. (2009) ADB: an efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks. *ACM Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Berkeley, California, Nov.4-6, pp. 43–56. ACM, New York, NY, USA.
- [24] Jiao, X., Lou, W., Ma, J., Cao, J., Wang, X., and Zhou, X. (2010) Duty-cycle-aware minimum latency broadcast scheduling in multi-hop wireless networks. *IEEE 30th International Conference on Distributed Computing Systems (ICDCS)*, Genoa, Italy, June 21-25, pp. 754–763. IEEE, New Brunswick, NJ USA.

- [25] Luo, S., Mao, X., Sun, Y., Ji, Y., and Tang, S. (2012) Delay minimum data collection in the low-duty-cycle wireless sensor networks. *IEEE Global Communications Conference (GLOBECOM)*, California, USA, Dec. 3-7, pp. 232-237. IEEE, New Brunswick, NJ USA.
- [26] Wang, X., Wang, X., Liu, L., and Xing, G. (2013) Dutycon: A dynamic duty-cycle control approach to end-to-end delay guarantees in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, **9**, 1-33.
- [27] Benson, J., O'Donovan, T., Roedig, U., and Sreenan, C. (2008) Opportunistic aggregation over duty cycled communications in wireless sensor networks. *ACM/IEEE Proceedings of the 7th international conference on Information processing in sensor networks (IPSN)*, Missouri, USA, April 22-24, pp. 307-318. ACM, New York, NY, USA.
- [28] Anastasi, G., Borgia, E., Conti, M., and Gregori, E. (2011) A hybrid adaptive protocol for reliable data delivery in wsns with multiple mobile sinks. *Oxford Journal: The Computer Journal*, **54**, 213-229.
- [29] Wang, F. and Liu, J. (2012) On reliable broadcast in low duty-cycle wireless sensor networks. *IEEE Transactions on Mobile Computing (TOMC)*, **11**, 767-779.
- [30] Nath, S. and Gibbons, P. (2007) Communicating via fireflies: Geographic routing on duty-cycled sensors. *ACM/IEEE Proceedings of the 6th international conference on Information processing in sensor networks (IPSN)*, Massachusetts, USA, April 25-27, pp. 440-449. ACM, New York, NY, USA.
- [31] Lee, S., Park, J., and Choi, L. (2007) AMAC: Traffic-Adaptive Sensor Network MAC Protocol through Variable Duty-Cycle Operations. *Proceedings of IEEE International Conference on Communications (ICC)*, Scotland, June 24-28. IEEE, New Brunswick, NJ USA.
- [32] Ye, W., Heidemann, J., and Estrin, D. (2004) Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, **12**, 493-506.
- [33] Van Dam, T. and Langendoen, K. (2003) An adaptive energy-efficient MAC protocol for wireless sensor networks. *ACM Proceedings of the 1th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Los Angeles, CA, USA, Nov.5-7, pp. 171-180. ACM, New York, NY, USA.
- [34] Yang, S., Tseng, H., Wu, E., and Chen, G. (2005) Utilization based duty cycle tuning MAC protocol for wireless sensor networks. *IEEE Global Telecommunications Conference (GLOBECOM)*, St. Louis, MO, USA, Nov.28- Dec.2, pp. 1-5. IEEE, New Brunswick, NJ USA.
- [35] Zheng, T., Radhakrishnan, S., and Sarangan, V. (2005) PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks. *19th IEEE Proceedings of International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1-8. IEEE, New Brunswick, NJ USA.
- [36] Wang, H., Zhang, X., and Khokhar, A. (2006) An energy-efficient low-latency MAC protocol for wireless sensor networks. *IEEE Global Telecommunications Conference (GLOBECOM)*, San Francisco, California, USA, Nov.27-Dec.1, pp. 1-5. IEEE, New Brunswick, NJ USA.
- [37] Lu, G., Krishnamachari, B., and Raghavendra, C. (2004) An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. *Proceedings 18th International Parallel and Distributed Processing Symposium (IPDPS)*, Santa Fe, New Mexico, USA, April 26-30, pp. 216-224. IEEE, New Brunswick, NJ USA.
- [38] El-Hoiydi, A. and Decotignie, J. (2004) WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. *IEEE Ninth International Symposium on Computers and Communications (ISCC)*, Alexandria, EGYPT, June 28 - July 1, pp. 244-251. IEEE, New Brunswick, NJ USA.
- [39] Shi, X. and Stromberg, G. (2007) SyncWUF: An ultra low-power MAC protocol for wireless sensor networks. *IEEE Transactions on Mobile Computing (TOMC)*, **6**, 115-125.
- [40] Chang, Y., Jiang, J., Sheu, J., and Shih, H. (2008) ADCA: an asynchronous duty cycle adjustment MAC protocol for wireless sensor networks. *IEEE Global Communications Conference (GLOBECOM)*, New Orleans, LA, USA, No.30-Dec.4, pp. 1-5. IEEE, New Brunswick, NJ USA.
- [41] Buettner, M., Yee, G., Anderson, E., and Han, R. (2006) X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. *ACM Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Boulder, Colorado, USA, Oct.31-Nov.3, pp. 307-320. ACM, New York, NY, USA.
- [42] Sun, Y., Gurewitz, O., and Johnson, D. (2008) RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *ACM Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Raleigh, NC, USA, Nov. 5-7, pp. 1-14. ACM, New York, NY, USA.
- [43] Sha, M., Xing, G., Zhou, G., Liu, S., and Wang, X. (2009) C-mac: Model-driven concurrent medium access control for wireless sensor networks. *The 28th Annual IEEE International Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 19-25, pp. 1845-1853. IEEE, New Brunswick, NJ USA.
- [44] Kim, Y., Shin, H., and Cha, H. (2008) Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Missouri, USA, April 22-24, pp. 53-63. ACM, New York, NY, USA.